CHANGE-POINT DETECTION AND ESTIMATION OF PIECEWISE CONSTANT PARAMETERS USING SPARSE LINEAR REGRESSION

Girija Ramesan Karthik, Katthi Jaswanth Reddy

Electrical Engineering, Indian Institute of Science (IISc), Bengaluru-560012, India

ABSTRACT

The very notion of the parameters being piece-wise constant implies a sparse first order derivative. This property combined with the concept of Linear Regression (Sparse Linear Regression) is used to estimate the parameters of quasi-stationary signals like frequency hopping (FH) signals and speech. In FH signals, this helps in the robust estimation of carrier frequencies and hop instances. This is possible even in the presence of multiple FH signals. In case of speech signals, quasi-stationarity of speech is leveraged to estimate time-varying (TV) piece-wise constant auto-regressive (AR) coefficients. These AR coefficients can be used to compress the speech signal.

***Index Terms*—** frequency hopping, sparse linear regression, auto-regressive process, LASSO, SCAD, alternating direction method of multipliers, block coordinate descent

1. INTRODUCTION

We analyse the use Sparse Linear Regression to tackle two different problems, namely Parameter estimation of frequency hopping signals [1] and speech compression [2]. Here we assume a linear model. Hence, given a sequence of observations $Y = [y_1, y_2, \dots, y_N]$, we have

$$\mathbf{Y} = \mathbf{AX} + e \tag{1}$$

where $\mathbf{A}$ is known and $\mathbf{X}$ needs to be estimated. e is additive noise.

The most common way to obtain $\mathbf{X}$ is to minimize $\|\mathbf{Y} - \mathbf{AX}\|_2^2$. This is called least-squares method. It is a convex function and admits a closed form solution. However, in many cases some function of $\mathbf{X}$ is sparse. Exploiting this property of $\mathbf{X}$ can provide better solutions as we incorporate prior information about the representation $\mathbf{X}$ in the form of its sparsity.

So, the optimization problem that we will be dealing with in this project is

$$\arg\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{AX}\|_2^2 \quad \textbf{s.t.} \quad \|f(\mathbf{X})\|_0 < \eta \tag{2}$$

where $f(\mathbf{X})$ is some function of $\mathbf{X}$. This is called Sparse Linear Regression. This can be re-written in the Lagrangian form i.e,

$$\arg\min_{\mathbf{X}} \left\{ \|\mathbf{Y} - \mathbf{AX}\|_2^2 + \lambda \|f(\mathbf{X})\|_0 \right\} \tag{3}$$

where the relation between λ and η depends on the data. The parameter λ can be tweaked to induce the desired sparsity to the solution i.e, higher lambda favours sparser solutions and vice-versa.

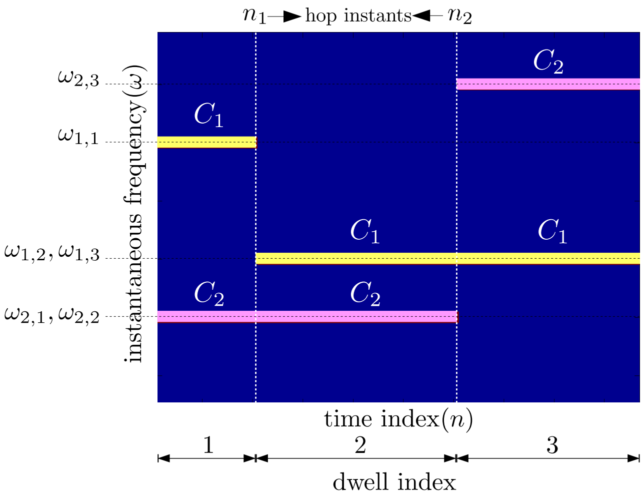

Fig. 1. Illustrative IF-gram of a signal with 2 FH signals. The IF-gram contains magnitude of the carrier frequencies at each time instant. The IF-gram contains 3 dwell periods and 2 hop instances n_1 and n_2. The m^{th} carrier frequency in the k^{th} dwell are shown to the left.

2. ESTIMATION OF FREQUENCY HOPPING SIGNALS

Frequency hopping is the phenomenon used in the transmission of radio signals where the carrier hops from one frequency to another. This was invented to prevent jamming of the signal. However, in the presence of multiple frequency hopping transmitters, a receiver receives all of them simultaneously. Hence, it is important to first extract accurate information about all the transmitted carriers and the hop instants so that the desired carrier can then be tapped into. Accurate hop instant estimation is very essential to prevent information loss.

2.1. Mathematical Model

Consider the noiseless signal $s(t)$, which at time $t \in [t_{k-1}, t_k)$ consists of M_k pure tones corresponding to the M_k frequency hopping signals i.e,

$$s(n) = \sum_{m=1}^{M_k} a_{m,k} e^{j2\pi\omega_{m,k}n}, \quad n_{k-1} \le n < n_k \tag{4}$$

where $n \in \{0, 1, \dots, N-1\}$, $a_{m,k} \in C$ is the complex amplitude and $\omega_{m,k} \in [-\pi, \pi]$ is the frequency of the m^{th} tone in the k^{th} system-wise dwell $[n_{k-1}, n_k)$. A system-wise dwell is an interval over which all tone frequencies and complex amplitudes remain constant. n_k is the k^{th} system-wise hopping instant and N is the

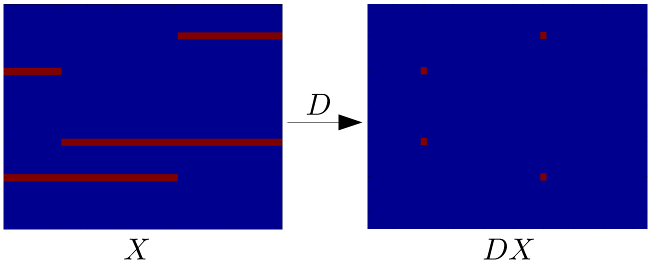

Fig. 2. The IF-gram X is sparse and so is the first order difference of X i.e, DX. This property can be used to introduce sparsity constraints into the objective function. (DX is the first-order difference operator.)

total duration of the signal. $s(n)$ is corrupted by additive circularly-symmetric complex Gaussian noise $v(n) \sim CN(0, \sigma^2)$ to give $y(n)$ i.e,

$$y(n) = s(n) + v(n), \quad 0 \leq n \leq N-1. \tag{5}$$

The matrix corresponding to the instantaneous frequency parameters similar to the spectrogram is shown in Figure 1. We refer to this as the instantaneous frequency gram (IF-gram). Let us define $\mathbf{Y} = [y_0, \ldots, y_{N-1}]^T$ and $\mathbf{W} = [w_0, \ldots, w_{N-1}]^T$, $w_i = [\mathbf{0}_{1\times Pi}, e^{j2\pi\omega_1 i}, \ldots, e^{j2\pi\omega_P i}, \mathbf{0}_{1\times(N-i-1)P}]^T$. We then define $\mathbf{X} = [A_0, A_1, \ldots, A_{N-1}]^T$ where A_i is the transpose of the i^{th} column of the IF-gram. Assuming no noise condition, we can now introduce the linear model for the FH signal as

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \tag{6}$$

2.2. Parameter estimation

Figure 2 shows the reference IF-gram considered for simulations in this project. The vectorized form of the IF-gram, X has been defined as $\mathbf{X}$. It can be seen from Figure that $\mathbf{X}$ is sparse. So, in line with equation 3 we can introduce $\|\mathbf{X}\|_0$ to obtain the Lagrangian as

$$\arg\min_{\mathbf{X}} \left\{ \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_2^2 + \lambda_1 \|\mathbf{X}\|_0 \right\} \tag{7}$$

Now, as seen in Figure 2, the first order difference of the columns of the of the IF-gram is also sparse. In fact this is sparser than the IF-gram. Incorporating this into equation 7 we get

$$\arg\min_{\mathbf{X}} \left\{ \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_2^2 + \lambda_1 \|\mathbf{X}\|_0 + \lambda_2 \|\mathbf{D}\mathbf{X}\|_0 \right\} \tag{8}$$

where $\mathbf{D}$ is the first-order difference operator modified to operate on the vectorized form, $\mathbf{X}$, of the IF-gram. We have now formulated the objective function to estimate the frequency-hopping parameters. However, the above objective function is non-convex and optimization is an NP-hard problem. So we introduce l_1 relaxation to make the function convex i.e,

$$\arg\min_{\mathbf{X}} \left\{ \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_2^2 + \lambda_1 \|\mathbf{X}\|_1 + \lambda_2 \|\mathbf{D}\mathbf{X}\|_1 \right\} \tag{9}$$

This objective function, though convex, does not admit to a closed form solution. However, there are several iterative methods to optimize this function. They include interior point solvers and coordinate descent algorithms. However, in large size problems, interior point solvers are not suitable and coordinate descent algorithms have issues converging to the global minima [1]. Angelosante et al. [1] derive update rules using alternating direction method of multipliers (ADMoM). The optimization problem is formulated as follows

$$[\hat{\mathbf{X}}, \hat{z}, \hat{u}] := \arg\min_{\mathbf{X},z,u} \left[\frac{1}{2}\|y - \mathbf{W}\mathbf{X}\|_2^2 + \lambda_1 \|z\|_1 + \lambda_1 \|u\|_1 \right] \\ \text{s.t. } z = \mathbf{X}, u = \mathbf{D}\mathbf{X} \tag{10}$$

The corresponding quadratically augmented Lagrangian is

$$\mathcal{L}(\mathbf{X}, z, u, \zeta, \mu) = \frac{1}{2}\|y - \mathbf{W}\mathbf{X}\|_2^2 + \lambda_1\|z\|_1 + +\lambda_1\|u\|_1 + \\ \mathcal{R}\{\zeta^H(x - z) + \mu^H(\mathbf{D}\mathbf{X} - u)\} + \\ \frac{c}{2}\left(\|x - z\|_2^2 + \|\mathbf{D}\mathbf{X} - u\|_2^2 \right) \tag{11}$$

The update rules are obtained using ADoMM are as follows.

$$\mathbf{X}^{(i)} = \frac{1}{c}(\mathbf{W}^H\mathbf{W} + \mathbf{D}^H\mathbf{D} + \mathbf{I}_{NP})^{-1}(c\mathbf{W}^H y + \\ cz^{(i-1)} + c\mathbf{D}^H u^{(i-1)} - \zeta^{(i-1)} - \mathbf{D}^H \mu^{(i-1)} - \mathbf{W}^H \rho^{(i-1)}) \tag{12}$$

$$z^{(i-1)} = \text{shrink}\left(\mathbf{X}^{(i)} + \frac{\zeta^{(i-1)}}{c}, \frac{\lambda_1}{c} \right) \tag{13}$$

$$u^{(i-1)} = \text{shrink}\left(\mathbf{D}\mathbf{X}^{(i)} + \frac{\mu^{(i-1)}}{c}, \frac{\lambda_2}{c} \right) \tag{14}$$

$$\zeta^{(i)} = \zeta^{(i-1)} + c\big(\mathbf{X}^{(i)} - z^{(i)}\big) \tag{15}$$

$$\mu^{(i)} = \mu^{(i-1)} + c\big(\mathbf{D}\mathbf{X}^{(i)} - u^{(i)}\big) \tag{16}$$

$$\text{shrink}(x, y) := \begin{cases} 0, & \text{if x = 0} \\ \frac{x}{|x|}\max\big(|x| - y, 0\big), & \text{otherwise} \end{cases} \tag{17}$$

2.3. Tuning regularization parameters

We know that the sparsity of $\mathbf{X}$ is directly proportional to λ_1. This implies that, with increase in λ_1, $\mathbf{X}$ gets closer to the zero matrix. Proposition 1 provides an upper bound on λ_1 to prevent this spurious solution.

Proposition 1: If $\lambda_2 = 0$, then $\hat{\mathbf{x}} = \mathbf{0}_{NP}$ if and only if $\lambda_1 \geq \lambda_1^* := \|\mathbf{W}^H\mathbf{Y}\|_\infty$

So, choosing $\lambda_1 < \lambda_1^*$ guarantees a non-zero solution to the optimization problem. The reference IF-gram gives a value of $\lambda_1^* = 2$. So, choosing $\lambda_1 > 2$ should ensure that we do not get the all zero solution. We will validate this proposition further in the next section. Let, T_N^l denote the $N \times N$ lower triangular matrix with all nonzero entries equal to one. Define $\Sigma :=\mathrm{T}_N^l \otimes I_P$, and partition the matrix product $W\Sigma$ into $M_0 \in \mathbf{C}^{N\times P}$ and $M \in \mathbf{C}^{N\times(N-1)P}$, so that $[M_0, M] := W\Sigma$.

Proposition 2: If $\lambda_2 = 0$ and M_0 has full column rank, then $\hat{\mathbf{x}} = [\mathbf{x}_c^T, \ldots, \mathbf{x}_c^T]^T$ with $\mathbf{x}_c^T := (M_0^H M_0)^{-1} M_0^H \mathbf{Y}$, if and only if $\lambda_2 \geq \lambda_2^* := \|M_0^H(M_0\mathbf{x}_c - Y)\|_\infty$

Similarly, the sparsity of $\mathbf{DX}$ is directly proportional to λ_2 and Proposition 2 provides an upper bound on λ_2. Choosing $\lambda_2 < \lambda_2^*$ ensures that $\mathbf{DX}$ is a non-zero vector.

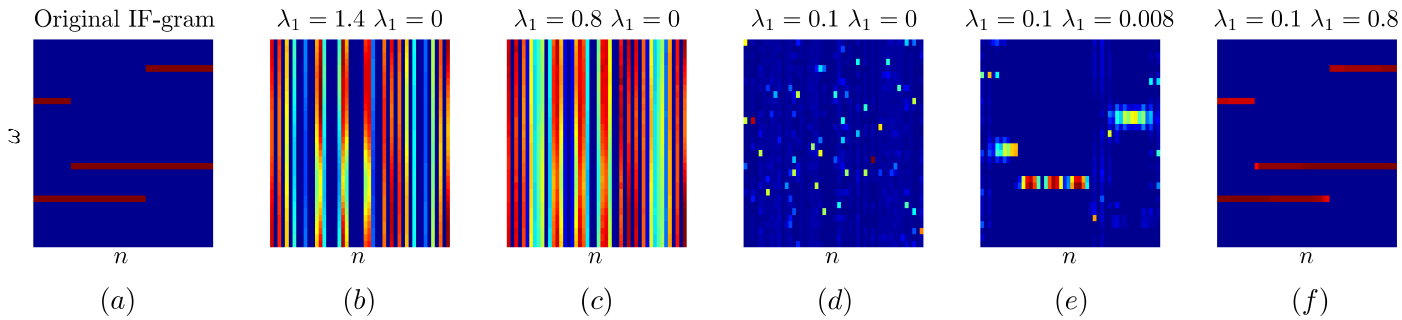

Fig. 3. The original IF-gram and the estimated IF-grams obtained for various values of λ_1 and λ_2. It can be seen that the estimated IF-gram gets progressively better as we move from (a) to (f)

2.4. Experiments and Results

2.4.1. *Validating the need for sparsity*

We first perform the experiment by considering λ_1 & $\lambda_2 = 0$. This is equivalent to least squares estimation. The estimated IF-gram is shown in Figure 4. It can be seen that the estimated IF-gram is dense and it does not match the original IF-gram.

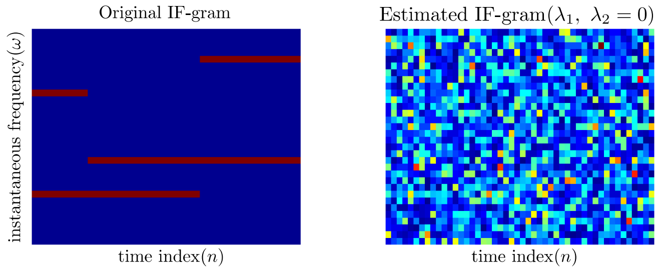

Fig. 4. For λ_1 and λ_2 = 0, it can be seen that the estimated IF-gram is very dense and does not correspond to the original IF-gram.

2.4.2. *Introducing λ_1*

Now, we consider regularization with $\lambda_1 > 0$ & $\lambda_2 = 0$. This introduces sparsity only on $\mathbf{X}$ and not on $\mathbf{DX}$. The estimated IF-grams for 3 different λ_1 values are shown in Figure 3(b-d). $\lambda_1 = 0.1$ gave the sparsest solution. However, the estimated IF-grams still do not match to the original. Note that all the λ_1 values are less than $\lambda_1^* = 2$ and all the estimated IF-grams are non-zero.

2.4.3. *Introducing λ_1 & λ_2*

Further, we consider both λ_1 & $\lambda_2 > 0$. We start with $\lambda_1 = 0.1$ and a low value for $\lambda_2 = 0.008$. It can be seen in Figure 3 that estimated IF-gram is far superior to the ones obtained without regularization using λ_2. For $\lambda_1 = 0.1$ and $\lambda_2 = 0.8$, the estimated IF-gram matches almost perfectly to the original IF-gram.

2.4.4. *Validating Proposition 1*

To validate proposition 1, we run the ADoMM updates for various values of λ_1 varying from 0.2 to 3 in steps of 0.2. The λ_1^* obtained for this particular signal is 2. So, under the assumption that ADoMM necessarily converges to the solution of equation , we expect $\|\mathbf{X}\|_1$ to be very close to 0 when $\lambda_1 \geq 2$.. As seen in Figure 5, $\|\mathbf{X}\|_1 \simeq 0$ when $\lambda_1 \geq 2$. It also validates that choosing $\lambda_1 < \lambda_1^* = 2$ ensures that X is non-zero.

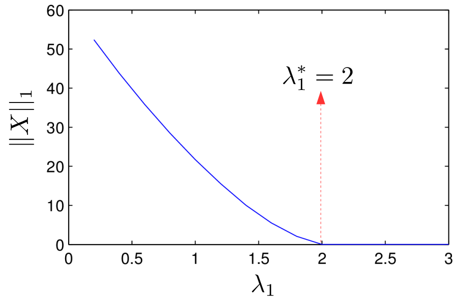

Fig. 5. l_1 norm of estimated IF-gram vs. λ_1

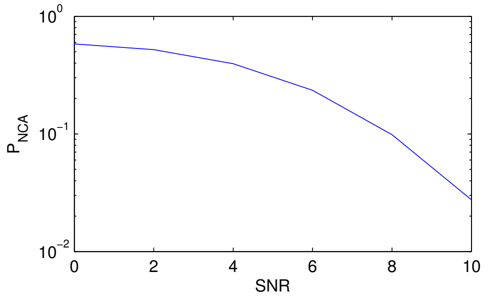

Fig. 6. Probability of incorrect hop instant estimation vs. SNR.

2.4.5. *Hop instant estimation under noisy conditions*

Let $[\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{N-1}]$ be the estimated IF-gram where $\hat{x}_i$ is the column corresponding to the i^{th} time index. Let $[d_1, d_2, \ldots, d_{N-1}]$ be the first-order difference matrix where $d_i = \hat{x}_i - \hat{x}_{i-1}$. We then calculate the vector $d = [\|d_1\|_1, \|d_2\|_1, \ldots, \|d_{N-1}\|_1]$. Under the assumption that the IF-gram is correctly estimated, d will be a sparse vector and the non-zero locations correspond to the hop instants. Therefore hop instants are estimated using the peaks of d.

We assume that the number of hop instances in known, i.e. in our case it is equal to 2. So we pick the 2 most prominent peaks from d and use their corresponding locations as the hop instants. The hop instant is determined as correctly estimated if the location lies within N_w instants from the actual location. We choose N_w = 3. Additive circularly-symmetric complex Gaussian noise is added to simulate SNRs from 0 to 10 in the steps of 2. We define P_NCA as the probability that the estimated hop instants were incorrect. This probability is calculated for each SNR by using 1000 different realizations of noise. Figure 6 shows the P_{NCA} vs. SNR curve.

3. TIME VARYING AR PARAMETER ESTIMATION

Sum-of-exponentials models with piecewise constant parameters, as discussed in the previous section, are encountered in several branches of engineering like communications and radar. But, the natural signals such as speech and electroencephalogram (EEG) do not conform with it. For signals like that, Autoregressive (AR) models have been deployed for parametric spectral estimation, since they form a dense set in the class of continuous spectra. While AR modeling of stationary random processes is well appreciated, many signals encountered in real life are nonstationary (e.g., speech signals). In the next section, TV-AR (Time Varying - Autoregressive) models with piecewise-constant coefficients are introduced. Identification of the coefficients is regarded as a sparse linear regression with grouped variables, which enables the usage of efficient algorithms.

3.1. Problem Statement

Let $\{y_n\}_{n=-L}^{N}$ denote the realization of an L^{th} order TV-AR process obeying the discrete-time input-output relationship as:

$$y_n = \sum_{l=1}^{L} a_{l,n} y_{n-l} + v_n, n = 0, 1, ..., N$$

where v_n denotes the zero-mean white input noise at time n with variance $\sigma^2 := E[v_n^2] < \infty$ and $a_{l,n} :=$ is the l^{th} TV-AR coefficient at time n. With $h_n := [y_{n-1}, y_{n-2}, ..., y_{n-L}]^T \in R^L$ and $a_n := [a_{1,n}, a_{2,n}, ..., a_{L,n}]^T \in R^L$, the observation model can be rewritten as

$$y_n = h_n^T a_n + v_n, n = 0, 1, ..., N \tag{18}$$

Suppose that abrupt changes in the spectrum of $\{y_n\}$ occur due to piecewise-constant changes of a_k, i.e.,

$$a_n = a_k, n_k \leq n \leq n_{k+1} - 1 \tag{19}$$

for k = 0,1,....,K, where K denotes the number of abrupt changes in the TV-AR spectrum, and n_k the time instant of the k^{th} abrupt change. The interval $[n_k, n_{k+1}-1]$ is referred to as the $k^{th} segment$. Without loss of generality, let n_0 = 0 and $n_{K+1} - 1 = N$.

In this context, the goal is to identify the instants $\{n_k\}_{k=1}^{K}$ where the given time series $\{y_n\}$ is split into $K + 1$ (stationary) segments, and also estimate the constant AR coefficients per segment, i.e., $\{a_k\}_{k=0}^{K}$. The number of abrupt changes, specifically K, is not necessarily known.

3.2. Parameters estimation

3.2.1. *Formulation of the problem statement*

Let, d_n denote the difference vector defined as:

$$d_n := \begin{cases} a_n, & \text{if } n = 0. \\ a_n - a_{n-1}, & \text{otherwise.} \end{cases} \tag{20}$$

Regularized LS is the approach used for estimating the parameters. With μ denoting a positive tuning constant, an l_0-type regularization is typically adopted to estimate the change points and the AR coefficients as

$$\{\breve{a}_n\}_{n=0}^{N} := \arg \min_{\{a_n\}_{n=0}^{N}} \left[\frac{1}{2} \sum_{n=0}^{N} (y_n - h_n^T a_n)^2 + \mu \sum_{n=1}^{N} \delta_{0_L}(d_n)\right] \tag{21}$$

where the indicator function $\delta_{0_L}(\cdot) : R^L \rightarrow 0, 1$ is defined as

$$\delta_{0_L}(a) := \begin{cases} 0, & \text{if } a = 0_L. \\ 1, & \text{otherwise.} \end{cases} \tag{22}$$

3.2.2. *Utilizing the Group Sparsity of Difference Vectors*

From a practical point of view, the minimization of (21) is challenging since an exhaustive search over all the possible sets of change instants has to be performed. Several techniques have been developed to evaluate it, out of which DP is one which solves the problem in polynomial time, the computational complexity is cubic in N, which limits their applicability to signal segmentation in practice since N can be very large in typical real time applications and cubic complexity cannot be afforded.

Hence, the convex relaxation of the cost in (21) is advocated based on recent advances in sparse linear regression and compressive sampling. To this end, (21) is first reformulated into a sparse regression problem with non-convex regularization that is subsequently relaxed through a tight convex approximation. It enables remarkably accurate retrieval of change points, obtained via an efficient ”block-coordinate descent iteration” that incurs only ”linear” computational burden and memory storage.

Therefore, now exploiting the group sparsity of coefficient changes, to disclose the connections between ”TV-AR signal segmentation” and ”sparse linear regression”, we will make use of d_n, the difference vectors.

Using (20), the problem in (21) can be rewritten as:

$$\{\check{d}_n\}_{n=0}^{N} := \arg \min_{\{d_n\}_{n=0}^{N}} \left[\frac{1}{2} \|y - Xd\|_2^2 + \mu \sum_{n=1}^{N} \delta_{0_L}(dn)\right] \tag{23}$$

where $d := [d_0^T, d_1^T, ..., d_N^T]^T \in R^{(N+1)L}$, and

$$X := \begin{pmatrix} h_0^T & 0_L^T & 0_L^T & \cdots & 0_L^T & 0_L^T \\ h_0^T & h_0^T & 0_L^T & \cdots & 0_L^T & 0_L^T \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{N-1}^T & h_{N-1}^T & h_{N-1}^T & \cdots & h_{N-1}^T & 0_L^T \\ h_N^T & h_N^T & h_N^T & \cdots & h_{N-1}^T & h_N^T \end{pmatrix} \tag{24}$$

And we can recover the coefficients $\{a_n\}_{n=0}^{N}$ easily from the difference vectors $\{d_n\}_{n=0}^{N}$ using

$$a_n = \sum_{n'=0}^{n} d_{n'} \tag{25}$$

3.2.3. *Introducing the convexity*

The non-convex regularization term in (23) makes it challenging. It ”pushes” most of the $\{d_n\}_{n=1}^{N}$ vectors towards 0_L, while d_0 is not penalized. As a consequence, the vector $\check{d} := [\check{d}_0^T, \check{d}_1^T, ..., \check{d}_N^T]^T$ is group sparse, and the nonzero group indices correspond to the change instants of the TV-AR coefficients.

Recently, a convex model selector with grouped variables was put forth by [3] and successfully applied to bio-statistics and compressive sampling. It generalizes the (nongrouped) Least-Absolute Shrinkage and Selection Operator (Lasso) [4] to regression problems where the unknown vector exhibits sparsity in groups; hence, its name ”group Lasso”. The crux of group Lasso is to relax the regularization in (23) with a tight convex approximation.

For grouped variables, it holds that the equivalent of ”the sparsity-promoting l_1-norm” is ”the sum of the l_2-norms” [3].

After group Lasso is applied for catching change points by estimating the difference vectors, (23) becomes:

$$\{\hat{d_n}\}_{n=0}^{N} := \arg \min_{\{d_n\}_{n=0}^{N}} \left[\frac{1}{2}\|y - Xd\|_2^2 + \lambda \sum_{n=1}^{N} \|d_n\|_2\right] \tag{26}$$

where λ is a positive tuning parameter.

It is known that the group Lasso regularization encourages group sparsity i.e., $\check{d}_n = 0_L$ for most $n > 0$ [3]. Again, larger the λ, the sparser the $\check{d}$.

$\|d_n\|_2$ is non-differentiable at $d_n = 0_L$, which enables group Lasso to encourage group sparsity.

Needless to say that convexity of the regularizing functions avoids the presence of local minima, and allows for solving the resulting optimization problem efficiently. To this end, an efficient ”block-coordinate descent algorithm” is developed in [2], with computational complexity per iteration that scales linearly with N.

3.2.4. Block-coordinate Descent Algorithm

The crux of block-coordinate descent is to iterate minimization of the function of interest over a group of variables, while keeping the rest fixed. We can write the objective function as:

$$J(d) := \frac{1}{2}\|y - Xd\|_2^2 + \lambda \sum_{n=1}^{N} \|d_n\|_2 \tag{27}$$

and let $d^{(i-1)} := [d_0^{(i-1)^T}, d_1^{(i-1)^T}, ..., d_N^{(i-1)^T}]^T$ denotes the provisional solution at iteration $i-1$. The n_{th} step of the i_{th} block-coordinate descent iteration entails minimization of $J(d)$ only with respect to d_n, while retaining the provisional estimates at iteration $i-1$, namely $\{d_{n'}^{(i-1)}\}_{n'=n+1}^{N}$, and the newly updated blocks at iterations i, namely $\{d_{n'}^{(i)}\}_{n'=0}^{n-1}$. Thus, block-coordinate descent at the n^{th} step of the i^{th} iteration yields

$$d_n^{(i)} = \underset{d_n}{\operatorname{argmin}}\, J\Big(\Big[d_0^{(i)}, ..., d_{n-1}^{(i)}, d_n, d_{n+1}^{(i-1)}, ..., d_N^{(i-1)}\Big]\Big) \tag{28}$$

for $n = 0, 1, ..., N$ and $i > 0$. Skipping constant terms, (10) can be rewritten as:

$$J(d) = \frac{1}{2}d^T X^T X d - d^T X^T y + \lambda \sum_{n=1}^{N} \|d_n\|_2 \tag{29}$$

$$\implies J(d) = \frac{1}{2}d^T R d - d^T r + \lambda \sum_{n=1}^{N} \|d_n\|_2 \tag{30}$$

where $R := X^T X$, and $r := X^T y$. Upon defining $R_{n:n'} := \sum_{m=n}^{n'} h_m h_m^T$ and $r_{n:n'} := \sum_{m=n}^{n'} h_m y_m$ for $n' \geq n$, it holds that

$$R = \begin{pmatrix} R_{0:N} & R_{0:N} & \cdots & R_{N-1:N} & R_{N:N} \\ R_{1:N} & R_{1:N} & \cdots & R_{N-1:N} & R_{N:N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ R_{N-1:N} & R_{N-1:N} & \cdots & R_{N-1:N} & R_{N:N} \\ R_{N:N} & R_{N:N} & \cdots & R_{N:N} & R_{N:N} \end{pmatrix} \tag{31}$$

and

$$r = \begin{pmatrix} r_{0:N} \\ r_{1:N} \\ \vdots \\ r_{N-1:N} \\ r_{N:N} \end{pmatrix} \tag{32}$$

Now, if we define $g_{n(i)}$ as:

$$g_{n(i)} = R_{n:N}\left(\sum_{n'=0}^{n-1} d_{n'}^{(i)}\right) + \sum_{n'=n+1}^{N} R_{n':N} d_{n'}^{(i-1)} - r_{n:N} \tag{33}$$

To build the block-coordinate descent algorithm, we need to define few terms first:

$$c_n^{(i)} := \sum_{n'=0}^{n-1} d_{n'}^{(i)} \tag{34}$$

$$s_n^{(i)} := \sum_{n'=n+1}^{N} R_{n':N} d_{n'}^{(i-1)} \tag{35}$$

it follows from (33) that,

$$g_n^{(i)} = R_{n:N} c_n^{(i)} + s_n^{(i)} - r_{n:N} \tag{36}$$

which shows that evaluating $g_n^{(i)}$ requires only the vectors $c_n^{(i)}$ and $s_n^{(i)}$. Given $\{d_n^{(i-1)}\}_{n=0}^{N}$ from the $(i-1)^{th}$ iteration, and initialising $c_n^{(i)}$ and $s_n^{(i)}$ at $n = 0$ as $c_0^{(i)} = 0_L$ and $s_0^{(i)} = \sum_{n=1}^{N} R_{n:N} d_n^{(i-1)}$, it is possible to recursively evaluate $c_n^{(i)}$ and $s_n^{(i)}$ given $c_{n-1}^{(i)}$, $s_{n-1}^{(i)}$ and $d_{n-1}^{(i)}$ from step $n-1$ for $n > 0$ as

$$c_n^{(i)} = c_{n-1}^{(i)} + d_{n-1}^{(i)} \tag{37}$$

$$s_n^{(i)} = s_{n-1}^{(i)} - R_{n:N} d_{n'}^{(i-1)} \tag{38}$$

Since, only $\{R_{n:N}\}_{n=0}^{N}$ and $\{r_{n:N}\}_{n=0}^{N}$ are needed to implement the block-coordinate descent algorithm (for evaluating $g_n^{(i)}$, $c_n^{(i)}$ and $s_n^{(i)}$), X need not be stored. Thus, the memory storage and complexity to perform one block-coordinate descent iteration grow linearly with N. This attribute renders the block-coordinate descent appealing especially for large-size problems where other approaches tend to be expensive.

The Block-coordinate descent algorithm is:

```
Given $\{R_{n:N}, r_{n:N}\}_{n=0}^{N}$.
Initialize with $d_n^{(0)} = 0_L$ for $n = 1, 2, ...N$.
for $i > 0$ do
    for $n = 0, 1, ...., N$ do
        if n = 0 then
            $c_0^{(i)} = 0_L$
            $s_0^{(i)} = \sum_{n=1}^{N} R_{n:N} d_{n-1}^{(i-1)}$
    $g_0^{(i)} = s_0^{(i)} - r_{0:N}$
    $d_0^{(i)} = -R_{0:N}^{-1} g_0^{(i)}$
  else
    $c_n^{(i)} = c_{n-1}^{(i)} + d_{n-1}^{(i)}$
    $s_n^{(i)} = s_{n-1}^{(i)} - R_{n:N} d_{n'}^{(i-1)}$
    $g_n^{(i)} = R_{n:N} c_n^{(i)} + s_n^{(i)} - r_{n:N}$
    if $\|g_n^{(i)}\|_2 \leq \lambda$ then
      $d_n^{(i)} = 0_L$
    else
$d_n^{(i)}$ =argmin$_{d_n} \in R^L \left[\frac{1}{2} d_n^T R_{n:N} d_n + d_n^T g_n^{(i)} + \lambda \|d_n\|_2\right]$
```

3.3. Tuning regularization parameters

Selection of λ is a critical issue since larger λ's promote sparser solutions , which translate to fewer changes in TV-AR spectrum. However, larger λ's increase the estimator bias as well. If the number of changes are known a priori, λ can be tuned accordingly. But, in general, there are no analytic tools to automatically choose the "best" λ.

Bound λ Proposition for avoiding the trivial(change-free) solution.

Proposition: *If X_0 has full column rank, then $\hat{d} = [d_{0,c}^T, 0_L^T, ..., 0_L^T]^T$ with $d_{0,c} := (X_0^T X_0)^{-1} X_0^T y$, if and only if $\lambda \geq \lambda^*$:*

$$\max_{n=1,2,....,N} \|X_n^T (X_0 d_{0,c} - y)\|_2$$

Therefore, λ must be chosen strictly greater than λ^* to avoid the trivial(change-free) solution.

The regularized function we are trying to minimize is :

$$\arg \min_{\{d_n\}_{n=0}^N} \left[\frac{1}{2}\|y - Xd\|_2^2 + \sum_{n=1}^N p_\lambda(|d_n|)\right] \qquad (39)$$

3.3.1. LASSO

$$p_\lambda^{Lasso}(|d|) = \lambda |d|$$

Through [5], it is revealed that Lasso tends to detect a "cloud" of small change points around an actual change. Moreover, due to the bias introduced by the Lasso, once the change points are obtained, another step is required to re-estimate the mean within a segment. The Lasso regularization possesses continuity but the estimates are biased, because in addition to small, large-amplitude coefficients are "shrunk" too.

3.3.2. SCAD

The Schwarz-like regularization yields unbiased estimates, but the solution is not continuous with respect to y. Hence, small variations of y or λ may result in large variations of $\hat{d}$ (this happens when one is uncertain whether to set the coefficient to 0 or not). On the other hand, the Lasso regularization possesses continuity but the estimates are biased, because in addition to small, large-amplitude coefficients are "shrunk" too.

An estimator which reduces the bias of group Lasso and can afford a convergent optimization solver is required, with three properties of *Unbiasedness, Sparsity and Continuity*. The corresponding algorithm is based on iterative instantiations of weighted group Lasso, which is capable of enhancing the sparsity of the solution, and thus improving the precision of the detected change points.

To overcme these limitations, the following Smooth Clipped Absolute Deviation (SCAD) regularization can be used with a > 2 [6]

$$p_\lambda^{SCAD}(|d|) = \begin{cases} \lambda|d|, & \text{if } |d| \leq \lambda \\ -\frac{d^2 - 2|d|a\lambda + \lambda^2}{2(a-1)}, & \text{if } \lambda < |d| \leq a\lambda \\ \frac{\lambda^2}{2}(a+1), & |d| > a\lambda \end{cases} \qquad (40)$$

Therefore, now given a provisional estimate of $\{d_n\}_{n=0}^N$ at iteration j-1, namely $\{\hat{d}_n^{(j-1)}\}_{n=0}^N$, the iterated approximation would be:

$$\{\hat{\mathrm{d}}_n^{(j-1)}\}_{n=0}^N = \text{argmin}_{\{d_n\}_{n=0}^N} \left[\frac{1}{2}\|y - Xd\|_2^2 + \sum_{n=1}^N p_\lambda^{SCAD'}\left(\|\hat{d}_n^{(j-1)}\|_2\right)\|d_n\|_2\right] \qquad (41)$$

for $j = 1, ..., J$. Since, the weights, $p_\lambda^{SCAD'}\left(\|\hat{d}_n^{(j-1)}\|_2\right)$ are non negative constants, the cost in (41) is convex and can be minimized using the block-coordinate descent algorithm. The role of the weights is to "avoid penalizing terms that, most likely, are non-zero".

3.4. Results and Experiments

For implementing the above mentioned approaches, two signals were chosen :

- A synthetically generated signal :
  3 cosine signals of different frequencies were generated and concatenated in increasing order.
- A speech signal :
  A speech signal of utterance Oh my god sampled at sampling frequency 16000 kHz was taken.

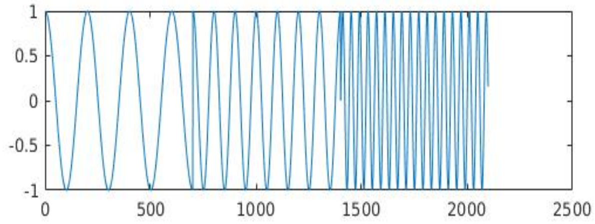

Fig. 7. Cosine Signal

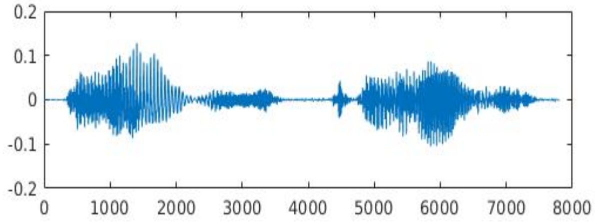

Fig. 8. Speech Signal

For the entire generated synthetic signal and a segment of the speech signal, the λ^* is obtained first. Then, ranging λ from 0 to $1.2\lambda^*$,TV-AR coefficients and the piece-wise constant segments are obtained. The obtained values are plotted and compared finally.

For the synthetic signal generated, the obtained plots for the range of λ's for both $Lasso$ and $SCAD$ regularization parameters are:

In each plot, the x-axis is the time axis and 8 TV-AR coefficients are plotted against it, where each of the 8 coefficients is plotted with different color.

In Figure 9, the TV-AR coefficients of order 8 for the synthetic signal generated are obtained for $\lambda = 0$, $\lambda^*/6$, $\lambda^*/2$ and $1.2\lambda^*$ respectively from top to bottom & the left image in each row is for $Lasso$ regularization, where as right image in each row is for $SCAD$ regularization.

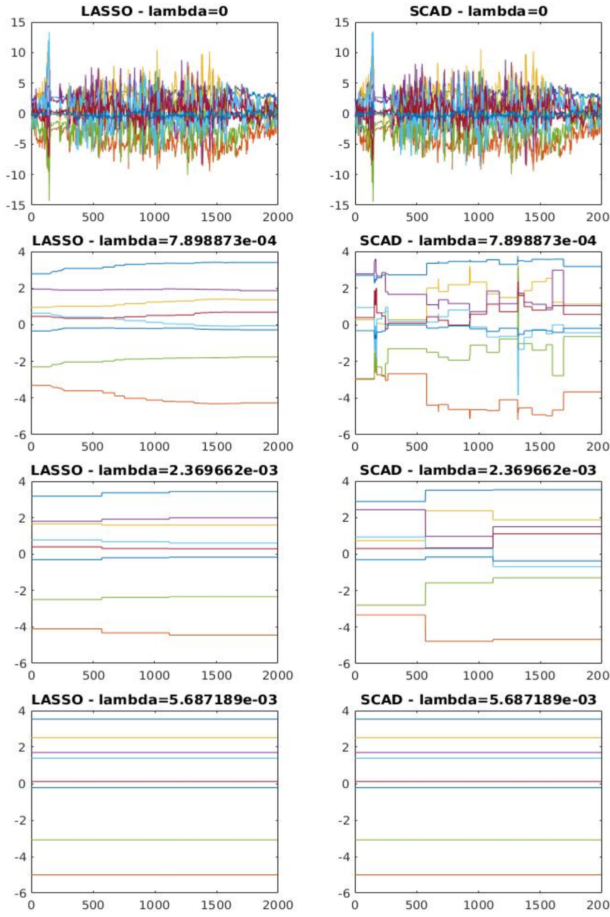

Fig. 9. 8^{th} order TV-AR coefficients of generated signal

In Figure 10, the TV-AR coefficients of order 8 for the part of the speech signal are obtained for $\lambda = 0$, $\lambda^*/6$, $\lambda^*/2$ and $1.2\lambda^*$ respectively from top to bottom & the left image in each row is for $Lasso$ regularization, where as right image in each row is for $SCAD$ regularization.

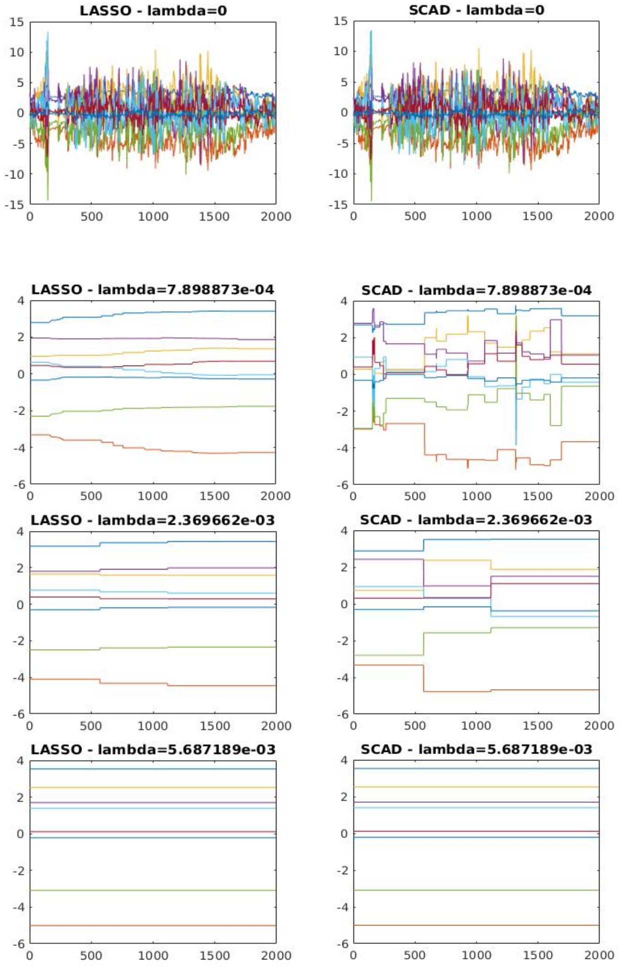

Fig. 10. 8^{th} order TV-AR coefficients of speech signal

To see how the group sparsity varies with the values of lambda, for the any signal, It has been tested over the generated synthetic signal and result is :

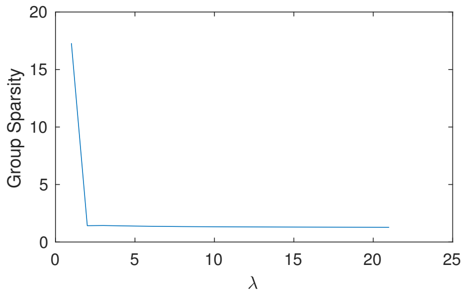

Fig. 11. λ versus Group Sparsity of the difference vectors

4. REFERENCES

[1] Daniele Angelosante, Georgios B Giannakis, and Nicholas D Sidiropoulos, "Estimating multiple frequency-hopping signal parameters via sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5044–5056, 2010.

[2] Daniele Angelosante and Georgios B Giannakis, "Group lassoing change-points in piecewise-constant ar processes," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 70, 2012.

[3] Ming Yuan and Yi Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[4] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[5] Zaid Harchaoui and Céline Lévy-Leduc, "Catching change-points with lasso.," in *NIPS*, 2007, vol. 617, p. 624.

[6] Jianqing Fan and Runze Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.